APPARATUS AND METHOD FOR IMPROVING RELIABILITY OF COLLECTED SENSOR DATA OVER A NETWORK

TECHNICAL FIELD

5   A field of the invention is sensor networks.

BACKGROUND ART

The convergence of techniques for sensing, communication, and processing has led to the emergence of wireless sensor networks. Recently, large-scale sensing has become feasible with the use of low-

10   cost, low-energy wireless sensor nodes. Many systems, for example in manufacturing, testing, and monitoring, collect data from a number of wireless sensors. The availability of these sensor networks enables sensing and monitoring of the physical world.

Even more so than in other applications that use wireless

15   data transfer, providing reliable data collection is a paramount concern in sensor networks, as the data is collected, processed, and used to make decisions in a machine-to-machine data collection framework. However, there are well-known problems with wireless data transfer

20   relating to the reliability and correction of data.

For example, a wireless network of sensor nodes is inherently exposed to various sources of unreliability, such as unreliable communication channels, node failures, malicious tampering of nodes, and eavesdropping. Sources of unreliability can be generally classified

25   into two categories: faults that change behavior permanently; and failures that lead to transient deviations from normal behavior, referred to herein as "soft failures".

Soft failures occur in wireless channels as transient errors, caused by noise at the receiver, channel interference, and/or multi-path fading effects. Additionally, the use of aggressive design technologies such as deep-sub-micron (DSM) and ultra-deep-sub-micron (UDSM) to
5   reduce the cost of each node further exposes the nodes to different types of transient errors in computations and sensing.

Most techniques for gauging reliability of sensor nodes place a high overhead on the collection. Typical existing reliability methods may add redundant hardware or transmit extra data at the
10   source to correct for data corrupted in the circuits or the communication channels respectively. This makes typical methods prohibitively expensive for use with heavily constrained sensor nodes. To address failures in circuits and communication channels, such methods incur high overheads in terms of energy budget, and design and
15   manufacturing cost in the sensor nodes.

Other prior methods for data correction include methods to correct soft failures in hardware as well as those to correct bit detection errors on a wireless communication channel. Techniques for correcting soft errors in hardware include both circuit-level and module-level
20   approaches, e.g. triple modular redundancy and error correction coding in hardware. Techniques for correcting bit detection errors on a wireless communication channel include parity-based forward error correction (FEC) coding techniques like channel coding, and retransmission-based techniques like ARQ.
25

DISCLOSURE OF THE INVENTION

Preferred embodiments of the present invention provide, among other things, an apparatus and method suitable for improving reliability of collected sensor data over a network. One or more
30   transient errors are predicted and corrected using correlation of

2

corrected data. For example, sensor data can be collected from one or more sensor nodes in a network. A device other than a sensor node can use the data to compute a predictive model based upon inherent redundancy in the data, and correct one or more later-received values

5   deemed unreliable.

Further features and advantages will become apparent from the following and more particular description of exemplary embodiments of the invention, and as illustrated in the accompanying drawings.

10

BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1 shows a network including a device used to perform a data aggregation and correction method according to a preferred embodiment of the present invention;

15   FIG. 2 schematically illustrates an algorithm implemented by an aggregator node for aggregating and correcting data from a data source, according to a preferred embodiment of the present invention;

FIG. 3 illustrates an exemplary operation for performing data correction, according to a preferred embodiment of the present

20   invention;

FIG. 4 shows an exemplary prediction history tree (PHT) for a delay of 3 samples, according to an exemplary embodiment of the present invention; and

FIG. 5 shows exemplary pseudo-code for implementing a

25   data aggregation and correction method, according to a preferred embodiment of the present invention.

BEST MODE OF CARRYING OUT THE INVENTION

Preferred embodiments of the invention provide improved reliability with minimal cost of error protection, i.e., the cost of sensor

30   nodes and communication overhead. In preferred embodiments, run-

time correction of transient errors originating either at the circuits of the sensor nodes or over the communication channel is conducted, with no design or operational overhead on the sensor node.

According to preferred embodiments of the invention, knowledge of the properties of the sensor data itself is used to achieve data checking and correction. Embodiments of the invention use information about correlations in sensor data, the goals of the sensor application, and its vulnerability to various errors.

For example, sensor data generally exhibits redundancy over a temporal period on a per-node basis, or over a cluster of nodes. Such inherent redundancy of the sensor data may be leveraged to make possible a high degree of reliability in data collection, without imposing overheads on sensor nodes, at the expense of nominal buffer requirements at data aggregator nodes, which are much less cost/energy constrained. Low-cost error correction apparatuses, systems, and methods for correcting soft failures according to preferred embodiments of the present invention are provided using the properties of data captured in a data prediction model.

Prior reliability techniques, by contrast, either added redundant hardware or transmitted extra data at the source to correct for data corrupted in the circuits or the communication channels, respectively. Such techniques are prohibitively expensive to be used with heavily constrained sensor nodes, and they do not use properties of the application data. Thus, to address failures in circuits and communication channels, these techniques incur prohibitively high overheads in terms of energy budget, and design and manufacturing cost in the sensor nodes.

An embodiment of the invention includes an application-level, data-aware method, for example implemented in software or encoded into a suitable device, for correction of transient errors in sensor data at an aggregation node, where aggregation and filtering of

the sensor data occur in a sensor data network. Preferred methods achieve run-time correction of data received from a data source, such as sensor nodes, over wireless communication channels, preferably without imposing any design or material cost, or performance overhead,

5   on the sensor nodes. Preferably, the overhead incurred is solely in terms of storage and computation costs at the data receiver, such as aggregator node(s) that buffer data for aggregation. The method preferably can be tuned to the performance requirements of the application and resource constraints at the aggregator.

10   Generally, a preferred method identifies and uses redundancies within the sensor data to correct the presence of transient errors. In exemplary embodiments, a detailed analysis of redundancy within sensor data captures correlation properties in a predictive model. The predictive model is then used during data acquisition for on-line

15   predictive correction of the data. This preferred method filters soft failures of the sensor data.

        More particularly, in exemplary embodiments a device, such as an aggregator node, develops a predictive model based on analysis of sensor data from sensor nodes of a network. The

20   aggregator node then conducts a reliability check at run-time using the predictive model to check for reliability of received data from the sensor nodes and to make error correction decisions. Preferred methods of the invention include collecting data offline for an inherent sensor data predictive model, and applying the model on-line at run-time.

25   While data predictions typically filter out the majority of errors in the observed values, it is possible that the predictions may not always track the data processes correctly. For example, aggregation operations performed by applications on collected data have varying levels of vulnerability to erroneous data. Preferred methods of the

30   present invention thus also delay the reporting of data within an application's delay constraints. The delayed reporting allows observed

values to be used in a preferably small set of later samples to guide the choice of corrected value between the predicted and observed value. Past data samples may also be used to help choose a corrected value. The preferred method can be tuned to the computational resources

5   available at the data receiver and the application's delay requirements by adjusting the delay.

A network embodiment of the invention includes one or more sensor nodes that wirelessly communicate data to one or more aggregator nodes. The inherent redundancy of the sensor data is

10   utilized to perform error correction at the site of data processing, which can, for example, be the aggregator node. This is beneficial, as the aggregator node typically has more computational, storage, and energy resources than the sensor nodes. Additional embodiments of the invention include an aggregator node configured for use in a wireless

15   network.

Referring now to the drawings, FIG. 1 shows a sensor network 10 that includes a device configured to perform an exemplary method according to the present invention. Preferably, the device is an aggregator node 12, which over a wireless channel 14 receives data

20   from a data source. The data source includes, for example, one or more sensor nodes 16, and preferably a plurality of sensors, which transmit data wirelessly via the channel 14. Preferably, the network 10 includes multiple aggregator nodes 12, though only one is shown in FIG. 1 for clarity.

25   The aggregator node 12 may include, for example, one or more modules for receiving and aggregating sensor data. The aggregations functions performed by these modules may include node-level or temporal aggregation 18 for aggregating data from a particular sensor and/or spatial or cluster-level aggregation 20, which aggregate

30   data from the different sensor nodes. Aggregated and corrected data

from the aggregator node 12 may in turn be sent to a server 22 or other device (i.e., reported) for processing or storage.

FIG. 2 shows a general schematic of an application-level algorithm for performing a data correction method according to
5   preferred embodiments of the present invention. The algorithm may be implemented in a device such as the aggregator node 18, for example, by any suitable method.

In an exemplary sensor data correction method, a predictive model of the data generation process is constructed,
10   preferably offline, by pre-processing of initially collected data (representative samples) from the sensor nodes 16. For example, suitable pre-processing logic, shown in FIG. 2 as a data model block 24, may be implemented in the aggregator node 12. This predictive model utilizes the correlation in the sensor data. Preferably, the correlation is
15   temporal, in which case the predictive model preferably is computed based on inherent temporal (per-node) redundancy in the sensor data. However, it is contemplated that other types of correlation may be additionally or alternatively used.

A model chosen should be rich enough for the predictions
20   to substantially match the data generation process. Also, the model should allow a prediction process that is efficient in terms of resource consumption and complexity to meet any performance requirements of the aggregator node 12, or other device. The choice of model generated by the data model block 24 given the above requirements
25   preferably will depend mostly on the level and nature of temporal correlation in the data. Though a variety of modeling techniques can be used to represent data correlation properties, the performance of the correction method largely depends on the accuracy of modeling and the efficiency of the predictions. An exemplary model, used in experiments
30   to test embodiments of the invention, is the auto-regressive moving average (ARMA) model. This is a linear predictive model that uses the

history of previous observations, shown in FIG. 2 as a data history block 26, as well as that of prediction performance, shown as an error history block 28. Order identification (that is, the number of past values and error history to be used for computing the new predicted value) for the

5   ARMA model may be performed by, for example, using the minimum final prediction error criterion.

As also shown in FIGs. 2 and 3, this predictive model is used at run-time for computing the likely value of the next reading, and the data correction method determines, based on the histories of

10 observed data and prediction errors, whether the value obtained from the sensor or that provided by the predictive model will be recorded or reported and used for future use. Put another way, the data correction may determine whether a value obtained by the sensor is reliable with respect to the likely value, and if not, it corrects or filters the value using

15 a predicted value.

This may be implemented, for example, via application-level predictive correction logic, shown in FIG. 2 as a data correction block 30. A preferred approach includes maintaining a history of observed data (data history block 26), and using the computed

20 predictive model to generate a predicted future value 32 from the history. After the next observed data value 34 is received from the sensor node 16, it is decided which of these candidate values to record. Preferably, the operation of the data correction block 30 is independent of the data model used for prediction. However, it is contemplated that the logic 30

25 for predictive correction may partially or fully overlap the logic 24 used for forming the predictive model.

In a general data correction method, as shown in FIG. 3, the data model block 24 of the aggregator node 12 wirelessly collects initial data from the sensor node 16 (step 40), processes the initial data

30 (step 42), and develops a predictive model (step 44) based on the processed initial data. During run-time operation the aggregator node

8

12 wirelessly receives and/or collects observed sensor data (step 46), and a likely value of the next reading from the sensor node 16 is predicted using the developed data model (step 48). Then, the data correction block 30 determines whether to use the received value (step

5   50), by determining the reliability of the received value. If the received value is reliable, this value is reported (step 52) as corrected data. If not, a transient error has been predicted by the aggregator node 12. In this case, the predicted value is reported (step 54) as corrected data to correct the transient error.

10   A significant issue in performing prediction-based correction is choosing how to handle mismatches between a predicted and observed value at the receiver (i.e., the aggregator node 12), which may have been caused by a genuine error or by departure of the data source's behavior from the model. Such errors should be handled

15   differently in these two cases. In preferred embodiments of the invention this decision is made based on past samples as well as a number of samples observed afterwards. This is performed using a delay, represented in FIG. 2 by a decision delay parameter (K) 58.

          Referring again to FIG. 2, Y represents the sequence of

20   observed values 34 of sensor data, Y' represents the results of a prediction block (predicted data 32), and Y_C represents the corrected values 60 from the data correction block 30. The data correction block 30 uses the predictive model developed by the data model block 24 in the process of correcting errors by generating and storing different

25   possible versions of the history of different predictions. At any point in time n, given observed data $Y(n)$ 34, the data correction block 30 computes the corrected value Y_C $(n-K)$ 60, where K represents the depth of the prediction history maintained for a posteriori correction.

          For example, and referring to FIG. 4, for a time n, the

30   observed values up to $Y(n)$ and the corresponding predictions up to $Y'(n)$ are used, after a delay of K samples, to report the corrected value

$Y_c(n-K)$ 60. For every sample of the sensor node 16 observed, the data correction block 30 compares it with the value predicted from the predictive model and past history, and attempts to report the value closer to the actual expected observation. The delaying of this decision

5   allows the step to consider the effect of any choice it makes on the prediction accuracy for the K samples following it.

In preferred embodiments, this delayed decision making is implemented using a prediction history tree (PHT) 70, which contains the possible predicted values and the corresponding prediction errors

10   for the past K samples. The prediction errors corresponding to each node's value in the PHT are stored in a parallel error history tree (not shown), which is maintained in sync with the PHT 70 by performing the same update operations on both trees.

An exemplary PHT 70 has a depth of K+1, and represents

15   the various potential values for the last K samples, i.e., $Y_c(i)$ where $i=n-K:n-1$. FIG. 4 shows an example of a PHT 70 for K=3. Each node 72 in any level j of the PHT 70 represents a possible value of $Y_c(n-K+j-1)$, with the root node (level 0) 74 denoting the value already chosen for $Y_c(n-K-1)$.

20   Every node has two outgoing paths 76, 78, labeled 0 and 1, respectively, in FIG. 4. These represent the choices of Y (observed value) and Y' (predicted value) respectively for the sample following it. Thus, every path from root to a leaf 80 in level $K+1$ denotes a series of up to 2^K choices leading to a sequence of values $Y_c(n-K:n-1)$. The

25   nodes 72 of the PHT in FIG. 4 are annotated with the possible values contained in them. For example, leaf node 82, annotated with $Y'(n-1|01)$, represents the predicted value Y'(n-1) obtained after following the path from the root node 74 through node 84 and node 86, corresponding to the choices of 011 from the root node.

30   Preferred methods use the PHT to select a value for forwarding to the server 22. An exemplary pseudo-code of a method

used to correct errors at the receiver using the PHT is shown in FIG. 5. At time n (step 90), observed value $Y(n)$ is received (step 92), and up to 2^K possible predicted values for that sample are computed, one for each path i (step 94) from the root to every leaf node. Each predicted value

5    $Y'(n, i)$ is computed (step 96) using a different set of data and error history based on the nodes on that path. Also, for every path, prediction error is computed (step 98), and the average prediction error per sample is computed (*PathErr*) using the prediction error (step 100). Based on the minimum path error, one of the child nodes of the root of

10   the PHT is selected (step 102) as the new root, and the content of the selected child node determines the corrected value of $Y_c(n-K)$ (steps 104, 106). The tree rooted at this child is then used to replace the PHT structure.

For example, the next-level PHT is generated (step 104).

15   In a preferred method for generating the PHT, the level 1 node (for example, node 84 in FIG. 4) is selected containing the path i (step 106). This node becomes node s. The observed and error values for node s are used for the corrected value Y_c and the prediction error reported to the application, as well as entered into the data and error history (step

20   108) The sub-tree rooted at the other branch from the root is discarded (step 110), and the remaining tree is extended another level (step 112) by adding one or two children (observed $Y(n)$ and prediction $Y'(n)$ for that path) to each leaf node.

To improve efficiency, the size of the prediction history (that

25   is, the PHT) can be somewhat reduced by assuming very small variations from the predictions to be due to randomness in the sensed physical process rather than transient errors.   As an exemplary implementation, an error threshold value ETH 114 may be used as a control parameter in a preferred method to avoid adding new $Y'(n)$

30   values if $E(n)$ is below ETH (step 116). This means that if that particular

11

leaf node becomes the root after N steps, the observed value Y should be used for Yc. Thus, the tree structure often will not be fully populated.

The choice of delay value K determines, apart from the delay in reporting the corrected values, the level of correction achieved

5   by the preferred data correction method under particular given data and error characteristics. The storage and computational complexity of the method also depend directly on the parameter K, since it determines the amount of history information used for correcting each sample. Since a preferred method distinguishes between modeling errors and real

10   random errors occurring in the sensor node 16 and/or the wireless channel 14, the optimum choice of K depends on the properties of the errors as well as the performance of the modeling technique used. Potentially, it is also possible to trade off correction accuracy against performance and resources by varying K, and match them to the

15   application requirements and constraints of the aggregator node 12.

The performance of a preferred correction method depends partly on the performance of the prediction algorithm. The prediction algorithm preferably is invoked for each path of every sample to predict the next value in that sequence. The primary resource consumed by the

20   correction block is storage, the space complexity being $O(2^K)$ for the PHT 70.

In these ways, for example, the delay may be tuned to a particular device, such as the aggregator node 12, or the wireless sensor network 10 by selection of K and by forming the PHT 70 based

25   on the selected K. Different depths of prediction histories may be used depending on the application's delay sensitivity, the relative error levels, and the resource constraints on the receiving node.

A number of methods, devices, and systems for data aggregation and correction have been shown and described, having

30   many features and advantages. By performing preferred data correction methods at the application level, design of a device or system

implementing the method can be made easier. By using the aggregator node 12 to perform data correction steps, overhead on the sensor nodes 16 is not increased, and computations can be performed using a device typically having far greater overhead. Use of a delay improves

5   the efficacy of a preferred method, and the delay can be chosen to tune the method to various devices or systems. An error threshold preferably reduces unnecessary overhead on the aggregator nodes 12.

Though various configurations of sensor networks are possible according to embodiments of the present invention, preferred

10   data aggregation and corrections methods are particularly useful within network architectures that include large numbers of cheap and light sensor nodes managed by aggregator nodes with comparatively larger energy and resource budgets.

While specific embodiments of the present invention have

15   been shown and described, it should be understood that other modifications, substitutions and alternatives are apparent to one of ordinary skill in the art. Such modifications, substitutions, and alternatives can be made without departing from the spirit and scope of the invention, which should be determined from the appended claims.

20   Various features of the present invention are set forth in the appended claims.